(19) Korean Intellectual Property Office (KR)

(12) Registered Patent Publication (B1)

(51) Int. Cl.[7]: G06F 17/00

(45) Examined Publication Date: March 24th 2005

(11) Registration No.: 10-0478857

(24) Registration date: March 16, 2005

(21) Application No.: 10-2002-0006646

(22) Application Date: February 06, 2002

(65) Unexamined publication No.: 10-2002-0036969

(43) Unexamined Publication Date: May 17, 2002

(73) Patentee: Safe I Co., Ltd.

306 Sungnam B/D, 1436-2 Onchon2-dong, Dongnea-gu Busan 607-837 S. Korea

(72) Inventors:

Ahn, Moo-Kyoung

Myeong Jang Sang-ga APT 306, Myeng Jang dong 145-9, Dongnae-gu Busan, S. Korea

Lee, Joo-Myeong

1403 Gangbyeon-hansin, Mapo-dong 350, Mapo-gu, Seoul, S. Korea

Sim, Kyeong-Sup

Woo 1 dong 618-34, Haeundae-gu, Busan. S. Korea

Im, Bum-chun

Amnam-dong 105 5 tong 6 ban, Seo-gu, Busan, S. Korea

Hwang, Myeong-guk

Dongdaesin 3-ga 61-4 1 tong 3 ban, Seo-gu, Busan, S. Korea
Jo, Hyeon-ho
Dongwoo artbilla A dong 302ho, Gwaebum-dong 555-37, Sasang-gu, Busan, S. Korea
(74) Agent: Gu, Sung-jin
Examiner: Choi, Young-hwan

(54) AUTOMATIC UPDATE SYSTEM USING ONLINE

[Summary of the Invention]

The present invention relates to an automatic update system, and more particularly, to a system automatically updating a database of a computer connected online without a user's administration.

The present invention comprises a network interface unit built in or connected to a client computer, so as to transmit a packet on request; a network device driver allowing an operating system to recognize the network interface unit and to operate the network interface unit; a query module included in or interlocked with the network device driver, so as to generate a packet for comparing a database version stored in a client to a database version stored in a server and transmit the packet to the network device driver; a comparison module installed in the server, so as to receive the data packet with a version information

transmitted from the client, and to compare the vaccine information to the database version stored in the server, and to provide the result to the client; an update module included in or interlocked with the network device driver so as to update the database stored in the client in accordance with the result of the comparison module of the server and an update database updated to a database with the same version as the database of the server by the update module.

According to the invention, a computer user can update a specific database file regardless of a setting of the computer.

[A Major Drawing]

Fig. 1

[INDEX]

Internet, update, database, network interface unit, network device driver

[Specification]

[Brief Description of the Drawings]

Fig. 1 is a configuration diagram illustrating an automatic update system using online according to an embodiment of the invention.

Fig. 2a is a flowchart illustrating an automatic update system using online according to an embodiment of the invention.

Fig. 2b is a flowchart illustrating an automatic update

system using online according to an embodiment of the invention.

Fig. 2c is a flowchart illustrating an automatic update system using online according to an embodiment of the invention.

<Reference Numerals>

A10: Client

B10: Server

C10: Application Program

100: Network Interface Unit

110: Flash Memory

111: Server information

112: Authentication value

113: Version Information

200: Network Device Driver

300: Query Module

400: Comparison Module

500: Update Module

600: Update Database

[Detailed Description of the Invention]

[Object of the Invention]

[Field of the Invention and Background Art]

The present invention relates to an automatic update system, more particularly, to a system updating a database of a computer connected online automatically without a

user's administration.

Generally In using a computer, users mainly use programs for users' convenience such as a vaccine program, a hacking-protect program, etc. The programs store a database concerning virus patterns or hacking patterns, and when the patterns stored in the database is detected, the programs perform operations such as cure or protect. Further, the programs are designed to operate on application programs and include functions to drive or stop by the user. However, the vaccine programs, the hacking-protect programs, or the like are installed, it is useless when an engine (or a pattern file, hereinafter referred to as a database) is not upgraded. That is, when new viruses appear, a new engine to cure the viruses is developed, but when the existing engines are not upgraded, the existing vaccine programs may not cure.

The database supporting the hacking-protect programs or the vaccine programs to protect user's computers are generally stored in auxiliary memory units and is operated according to corresponding programs.

In the vaccine programs, when users register, information about the newest vaccine is provided for the users, by using registered emails. The users visit homepages of the corresponding producer of the vaccine programs and receive databases concerning the newest vaccine. Finally, the users select the related update items and

receive the corresponding databases.

However, a problem of the above method is that the vaccines do not properly perform when the databases are not updated due to the users' business or other matters.

In addition, management menus the corresponding programs include sub-programs to periodically update the databases of the corresponding programs for the users to update the newest database. Further, in order to properly update the databases, menus related to the update of the sub-program have to be set. When the vaccine programs or the hacking-protect programs are active, the corresponding programs automatically connect servers providing vaccine or hacking-protect databases and download the database in accordance with setting periods.

However, since users have to set the function of the vaccine or hacking-protect programs provided in the sub-program, computer beginners may get difficulty of employing the programs

In addition, another problem of the above method is that the corresponding databases have to be repeatedly downloaded when computers are initialized due to program reinstallation, hard-disk format, etc.

[Field of the Invention and Background Art]

The present invention relates to an automatic update system, more particularly, to a system updating a database

of a computer connected online automatically without a user's administration.

Generally in using a computer, users mainly use programs for users' convenience such as a vaccine program, a hacking-protect program, etc. The programs store a database concerning virus patterns or hacking patterns, and when the patterns stored in the database is detected, the programs perform operations such as cure or protect. Further, the programs are designed to operate on application programs and include functions to drive or stop by the user. However, the vaccine programs, the hacking-protect programs, or the like are installed, it is useless when an engine (or a pattern file, hereinafter referred to as a database) is not upgraded. That is, when new viruses appear, a new engine to cure the viruses is developed, but when the existing engines are not upgraded, the existing vaccine programs may not cure. The database supporting the hacking-protect programs or the vaccine programs to protect user's computers are generally stored in auxiliary memory units and is operated according to corresponding programs.

In the vaccine programs, when users register, information about the newest vaccine is provided for the users, by using registered emails. The users visit homepages of the corresponding producer of the vaccine programs and receive databases concerning the newest vaccine.

Finally, the users select the related update items and receive the corresponding databases.

However, a problem of the above method is that the vaccines do not properly perform when the databases are not updated due to the users' business or other matters.

In addition, management menus the corresponding programs include sub-programs to periodically update the databases of the corresponding programs for the users to update the newest database. Further, in order to properly update the databases, menus related to the update of the sub-program have to be set. When the vaccine programs or the hacking-protect programs are active, the corresponding programs automatically connect servers providing vaccine or hacking-protect databases and download the database in accordance with setting periods.

However, since users have to set the function of the vaccine or hacking-protect programs provided in the sub-program, computer beginners may get difficulty of employing the programs

In addition, another problem of the above method is that the corresponding databases have to be repeatedly downloaded when computers are initialized due to program reinstallation, hard-disk format, etc.

[Technical Goals of the Invention]

The invention is made in order to provide solutions to

the problems, and an advantage is to provide the automatic update system without user's update administration.

Another advantage is to provide a system free from the hard disk format or reinstallation of users.

[Configuration and Operation of the Invention]

In order to achieve the advantages, the present invention comprises a network interface unit built in or connected to a client computer and transmitting a packet at an one's request; a network device driver recognizing the network interface unit on an operating system and operating the network interface unit; a query module included in or interlocked with the network device driver, generating a packet to compare a database version stored in a client to a database stored in a server, and transmitting to the network device driver; a comparative module installed to drive, receiving the data packet with a version information transmitted from the client, and comparing to the database version stored in the server to provide the result to the client; an update module included in or interlocked with the network device driver and updating the database stored in the client in accordance with the result provided from the comparison module, and an update database updated to a version database equal to the database of the server by the update module.

The network interface unit includes the flash memory

storing the version information of the client provided to the server by the query module.

The network interface unit includes the flash memory storing the update database provided from the server and updated by the update module.

The network interface unit includes the flash memory storing an authentication value of the client provided to the server by the query module.

Further, the query module communicates with the server and transmits data such as the authentication value of the client.

The query module the query module generates a packet to query to the server and transmits to the network device driver, after an operating system recognizes the network interface unit and drives the network device driver.

Further, when the client computer is booted and then an initial packet is transmitted out via the network device driver (200), the query module generates a packet to query to the server and transmits to the network device driver.

After the client computer is booted, the query module generates a packet to query to the server at regular intervals and transmits to the network device driver.

Further, the query module generates a packet to query to the server at every specific time of a specific day and transmits to the network device driver.

When the client computer ends, the update module writes the update database updated on the flash memory.

Further, the update module performs real-time writing on the flash memory when the update database is updated.

An automatic update system using online according to an embodiment of the present invention is described in detail below, referring to attached drawings. Fig. 1 is a diagram of the automatic update system using online according to the embodiment of the invention.

The automatic update system using online according to the embodiment has major parts: a network interface unit (100), a network interface driver (200), a query module (300), a comparison module (400), an update module (500), and an update database (600).

First, the network interface unit (100) is described. The network interface unit (100) is generally called a LAN card, but preferably referred to a hardware apparatus for connecting online. Three major standards of the LAN in the world are an Ethernet, a Token Ring, and an Arc net. Of the standards, the Ethernet is worldly used. It is estimated that approximate 90% of the nationally used LAN are the Ethernet. For this reason, the flash memory is often used in place of a portable hard disc or as a supplement. The most used example is a memory apparatus inserting in a PCMCIA slot of a portable computer. According to the

embodiment, the flash memory (110) stores an authentication value (112) preferably to communicate with a server, an update database (600) described below, a version information of the update database, a server information (111) such as an IP address of a query server (B10) and a domain address, etc. The authentication value (112) corresponds to a value to authenticate when the server (B10) communicates with a client (A10) and the update database (600) refers to a new update database when an initial database communicates with the server (B10) and receives an update. The version information (113) refers to an update data of the database stored in the flash memory (110), and the server information (111) indicates data about the server (B10) providing data about an update. The authentication value (112), the version information (113), and the server information (111) will be described in the query module respectively.

Next, the network device driver (200) is described. The network device driver (200) recognizes the network interface unit (100) on an operating system and operates the network interface unit (100). Practically, the network device driver (200) corresponds to a device driver to communicate with a LAN controller contained in the network interface unit (100) and operates in a kernel mode. The network device driver (200), for example, operates in the kernel mode in the Microsoft Window 9x or the Window NT

operating system. In particular, the Window NT uses a privileged mode or a non-privileged mode, which are generally called a kernel mode or a user mode. Components executed in the kernel mode directly accesses a hardware and a software resource. Further, the Window NT executes core parts in the kernel mode to guarantee security and stability of a system. A microkernel, HAL (hardware abstraction layer), Device Driver and the like are executed in the kernel mode.

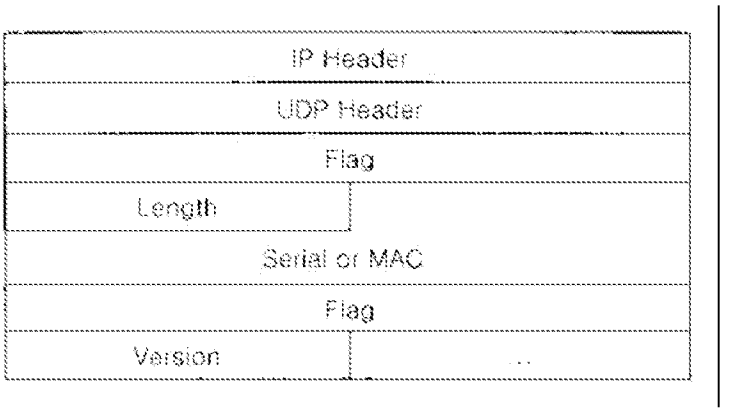
Accordingly, the network device driver (200) corresponds to the Device Driver. In addition, a program executed in the kernel mode is protected from offense by intention or by accident due to an own process design. An environmental sub-system such as Win32 sub-system, POSIX sub-system, etc and all application programs are executed in the user mode. In the user mode, a system user may break or stop execution of the corresponding program, while the user may not break or stop directly in the kernel mode. The kernel has responsibility of all system operations and almost all system functions. The Window NT uses the microkernel, which means that minimum required functions are contained.

The microkernel of the Window NT assigns many function traditionally executed in the kernel to an NT execution unit called part. The microkernel is a part of the NT execution

unit, and the NT execution unit is executed in the kernel mode. The network device driver (200) is configured to be driven the query module (300) and the update module (500) inside the network device driver (200). According to the OSI (Open System Information) reference model, the network device driver (200) is driven in the data link layer.

Next, the query module (300) is described. The query module (300) is included in or interlocked with the network device driver (200), and generates packets comparing the database version stored in the client to the database stored in the server, to thereby provide to the network device driver (200). That is, as described, the query module (300) operates in the kernel mode.

Table 1



The above table 1 is the packet generated by the query module (300) according to an embodiment of the invention. The packet is generated when the packet is demodulated or new-generated, as described in detail below.

When the client computer is booted and an initial packet is transmitted through the network device driver (200), a method of generating a query packet to the server (B10) is that the initially transmitted packet generated in TCP/IP Stack existing in a kernel is demodulated in the query module (300) or new-generated in the form suitable to TCP/IP specifications.

When the packet is demodulated, required data in IP Header and TCP Header of the packet generated in the TCP/IP Stack are extracted to modulate the IP Header and UDP Header separately. The reason the TCP Header is modulated to the UDP Header is that operation in the network device driver (200) is efficiently performed, and it is possible to generate the UDP Header itself. In addition, the packet generated in the TCP/IP Stack at a user's request is modulated to the UDP Header to refer to the corresponding destination address when it is necessary to check the destination address of the TCP Header data requested by the user

When the packet is new-generated, the IP Header and UDP Header are first generated, and the back of the Header includes required data. The above-mentioned required data are the version information (113) and the authentication value (112). Further, when the IP Header is generated, the query module (300) inserts the destination address referring

to the server information (111) stored in the flash memory (110).

A source address and a destination address are not present in the table 1, but the source address and the destination address exist in the IP Header, and the destination address is stored in the flash memory (110) of the network interface unit (100) to refer to the server information (111) or stored in the query module (300) to refer.

Next, when the query module (300) generates the packet to transmit to the server (B10) at regular intervals, or an operating system recognizes the network interface unit (100) to drive the device driver (200) at the time of booting, and then generates the query packet to transmit the server (B10), another method that is different from the method of modulating the packet transmitted from the TCP/IP Stack is used to generate a separate packet. The way to generate the packet is the same as the described way. After the IP Header and the UDP Header are generated, the query module (300) adds the authentication value (112) and the version information (113). The authentication value (112) has a serial or MAC address. The serial corresponds to the authentication value (112) and may use the MAC address as the authentication value. Generally, the MAC address is a unique number assigned to the network interface unit (100),

and the serial is made using another system different from the MAC address. The serial is stored in the flash memory (110) included in the network interface unit (100) and the query module (300) refers to the serial to generate.

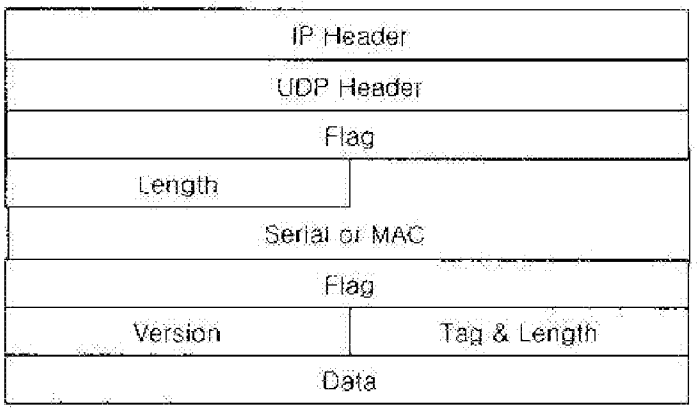
The version information (113) in the table 1 corresponds to Version and is transmitted to compare with the database version stored in the server. Further, the version information (113) may include a plural of database data as well as one database data. In addition, a Flag in the table 1 includes the detailed content of data area in the packet generated by the query module (300). That is, the Flag identifies the data transmitted between the server (B10) and the client (A10). The detailed description regarding the serial is omitted because of a general concept.

Next, the comparison module (400) is described. The comparison module (400) is installed in the server (B10) to drive, and receives the packet about the version information from the client (A10) to compare with the database version stored in the server (B10). Accordingly, the comparison module (400) has a role of providing the result. The comparison module (400) compares the version information provided from the client (A10) to the database version (600) stored in the server (B10) when the authentication value (112) that the query module (300) in the client (A10) generates and transmits is identified by the server (B10).

For example, when versions transmitted from the query module (300) are A Database: 1.0.2, B Database: 2.1.4, and C Database: 1.7.3, and the database versions stored in the server (B10) are A Database: 1.0.5, B Database: 2.2.3, and C Database 1.7.3, the comparison module (400) determines to update the A and B Database and not to update the C Database since the C Database version stored in the client (A10) is the same as the database version stored in the server (B10).

Accordingly, the comparison module (400) generates to provide the packet to the query module (300) of the client (A10) to update. The packet generated by the comparison module (400) is shown in table 2.

Table 2



Roles of Length and Serial or MAC are omitted since the roles are the same as that of the query module (300), and Version also corresponds to the database version information (113) stored in the server B (B10). The packet includes the

version information so that the version information (113) is provided from the client (A10) when the query module (300) of the client (A10) generates. Flag in the table 2 is required to determine whether the packet is transmitted to update contents stored in data area or to delete the part of the database or the entire database.

As the query module (300), the Flag is added, and the reason is described the above role of the query module (300).

Next, the update module (500) is described. The update module (500) is included in or interlocked with the network device driver (200) and updates the database stored in the client (A10) according to the result obtained in the comparison module (400) of the server (B10). The update module (500) reads data value corresponding to the Version of the entire packet transmitted from the comparison module (400) of the server (B10), and since the Version is the database version to update, the Version is separately stored in the flash memory (110) of the network interface unit (100). The encrypted database included in the data area in the table 2 is written to the flash memory (110). In addition, when the update module (500) updates a plural of the database, the update module (500) update versions of the corresponding database, respectively.

For example, when versions transmitted from the query module (300) are A Database: 1.0.2, B Database: 2.1.4, and C

Database: 1.7.3, and versions provided from the comparison module (400) are A Database: 1.0.5, B Database: 2.2.3, and C Database 1.7.3, the comparison module (400) determines to update the A and B Database and not to update the C Database since the C Database version stored in the client (A10) is the same as the database version stored in the server (B10).

In addition, value corresponding to the version included in the data area of the entire packet provided from the comparison module (400) is separately stored in the flash memory (110) to perform update query in the next time.

The update module (500) performs a write function to the flash memory (110) loaded in the network interface unit (100), and as a method of performing the write function, the method of recording the contents recorded in a flash image into the flash memory (110) is employed. The flash image corresponds to a memory of a kernel area, and it is the flash memory that makes the memory of the kernel area into a form stored in the flash memory. The reason to perform the above method is, for example, to perform a batch processing by converting some number of the flash images into one flash image, rather than to perform repeatedly at every request of writing when the real-time update is performed.

The update module (500) deletes the database as well as adding. The update module (500) performs the deletion when unnecessary parts or changed parts are made in the lists

stored in the database.

Next, the update database (600) is described. The update database (600) is updated to a new database version equal to the database of the server (B10) by the update module (500). Further, the update database (600) refers to a new database that the update module (500) adds or deletes from the stored database. As described in the query module (300), the update database (600) includes a plural of the database, rather than one database.

Referring to the attached Figs, the automatic update system using online according to an embodiment is described in detail below. Fig. 2a, Fig. 2b and Fig. 2c are flowcharts of the automatic update system using online according to the embodiment. First, A user boots a client (A10) (S100). In the middle of booting, an operating system recognizes a network interface unit (100) loaded in or connected to the client (A10), and the network device driver (200) is driven in the operating system (S102). At this time, the network device driver (200) is uploaded in the memory of the operating system (S103) and contents of a flash memory (110) built in the network interface unit (100) are uploaded in a kernel memory at the same time.

After ending the booting process, the user of the client (A10) performs Internet web browser (S106).

The performed web browser tries to request in order to

connect on the main page set as an application program by the user (S107). According to the request of connection, TCP/IP Stack of a kernel generates IP Header and TCP Header (S108) to provide to the network device driver (200) (S109), and therefore the query module (300) generates the IP Header first. Referring to server information (111) corresponding to a destination address in the IP Header from the contents of the flash memory (110), IP Header is generated (S111), and then UDP Header is generated (S112).

The query module (300) attaches Flag to identify data transmitting each other between a server (B10) and the client (A10) (S113), and attaches an authentication value (112) corresponding to Serial or MAC address stored in the flash memory (110) (S114). The query module (300) attaches database version information (113) stored in the flash memory (110) area (S115), and then attaches other required data to complete a packet (S116).

Next, the packet generated in the query module (300) is transmitted to the network device driver (200), and then the packet is transmitted from the network device driver (200) to the network interface unit (100) and the server (B10) (S118).

Therefore, the packet transmitted from the client (A10) is transmitted to the server (B10) (S119), the server (B10) authenticates the client (A10) by using the authentication

value (S120), and the server (B10) compares the version information (113) in the transmitted packet to the database version in the server (B10) to update (S121).

When the update is required (S122), a packet provided to the client (A10) is generated. First, referring to the packet provided to the client, the server generates the IP Header (S122-1) and the UDP Header (S122-2).

Afterwards, the Flag is attached (S122-3), and the authentication value (112) and the version information (113) stored in the flash memory (110) are attached (S122-4).

In addition, the update database (600) is attached to the data area of the packet (S122-5), and after the packet is completed (S122-6), the packet is transmitted to the client (A10) (S122-7). Even when the update is not required (S123), the packet is also generated to provide to the client (A10) in the same way. First, referring to the server information (111) stored in the flash memory (110), the server generates the IP Header (S123-1) and the UDP Header (S123-2). Afterwards, the Flag is attached (S123-3), and the authentication value (112) and the version information (113) stored in the flash memory (110) are attached (S123-4). After the packet is completed, the packet is transmitted to the client (A10) (S123-5).

The packet transmitted above is transmitted to the client (A10) (S124), and the version in the received packet

is compared to the database version information (113) stored in the flash memory (110). When the version in the transmitted packet is higher (S125), the version is written on the storage area of the version information (113) in the flash memory (110) (S125-1) and the database stored in the data area of the transmitted packet is written on the update database (600) in the flash memory (110) (S125-2).

The version in the received packet is compared to the version information (113) of the database stored in the flash memory (110). When the version in the transmitted packet is equal, the update module (500) is not performed.

[Effects of the Invention]

As described above, the automatic update system using online according to the embodiment has an advantage of the automatic update without a user's operation.

The automatic update system using online has another advantage of not requiring additional database update when a user formats or re-installs a computer.

(57) [Claims]

[Claim 1]

Deleted

[Claim 2]

An automatic update system using online, comprising:

a network interface unit built in or connected to a client computer, so as to transmit a packet on request;

a network device driver to allowing an operating system to recognize the network interface unit and to operate the network interface unit;

a query module included in or interlocked with the network device driver, so as to generate a packet for comparing a database version stored in a client to a database version stored in a server and transmit to the network device driver;

a comparison module installed to drive, so as to receive the data packet with a version information transmitted from the client and to compare to the database version stored in the server, and to provide the result to the client;

an update module included in or interlocked with the network device driver so as to update the database stored in the client in accordance with the result of the comparison module of the server;

an update database updated to a version information base equal to the database of the server by the update module; and

wherein the network interface unit includes a flash memory storing the update database and the version information of the update database provided from the server and updated by the update module.

[Claim 3]

Deleted

[Claim 4]

The automatic update system according to Claim 2, wherein the network interface unit includes the flash memory storing an authentication value of the client provided to the server by the query module.

[Claim 5]

Deleted

[Claim 6]

The automatic update system according to Claim 2, wherein the query module communicates with the server and transmits data such as the authentication value of the client.

[Claim 7]

The automatic update system according to Claim 2, wherein when the operating system recognizes the network interface unit at the time of booting and drives the network device driver, the query module generates a packet to be queried to the server and transmits the packet to the network device driver.

[Claim 8]

The automatic update system according to Claim 2, wherein when the client computer is booted and then an initial packet is transmitted through the network device driver (200), the query module generates a packet to query

to the server and transmits to the network device driver.

[Claim 9]

The automatic update system according to Claim 2, wherein after the client computer is booted, the query module generates a packet to be queried to the server at regular intervals and transmits the packet to the network device driver.

[Claim 10]

The automatic update system according to Claim 2, wherein the query module generates a packet to be queried to the server at a specific time of a specific day and transmits the packet to the network device driver.

[Claim 11]

The automatic update system according to Claim 2, wherein when the client computer ends, the update module writes the update database updated to the flash memory.

[Claim 12]

The automatic update system according to Claim 2, wherein the update module writes the update database to the flash memory in real time.

Drawings

A10 CLIENT

B10 SEVER

100 INTERNET INTERFACE DEVICE

110 FLASH MEMORY

200 NETWORK DEVICE DRIVER

300 QUERY MODULE

400 COMPARISON MODULE

500 UPDATE MODULE

600 UPDATE DATABASE

PROVIDE UPDATE

PERFORM UPDATE

REFERENCE

Fig. 2a

START

S100 BOOT CLIENT

S101 RECOGNIZE NETWORK INTERFACE UNIT

S102 DRIVE NETWORK DEVICE DRIVERS

S103 UPLOAD IN OS MEMORY

S104 UPLOAD CONTENT OF FLASH

S105 END BOOTING

S106 DRIVE WEB BROWSER

S107 REQUEST TO CONNECT ON MAIN PAGE

S108 GENERATE IP/TCP HEADER

S109 PROVIDE NETWORK DEVICE DRIVER

END

Fig. 2c

START

S120 PERFORM AUTHENTICATION

S121 DETERMINE TO UPDATE OR NOT

S122 UPDATE?

YES

NO

S122-1 GENERATE IP HEADER

S122-2 GENERATE UDP HEADER

S122-3 ATTACH FLAG

S122-4 ATTACH AUTHENTICATION VALUE & VERSION INFORMATION

S122-5 ATTACH DATABASE TO UPDATE

S122-6 COMPLETE PACKET

S122-7 TRANSMIT TO CLIENT

S123-1 GENERATE IP HEADER

S123-2 GENERATE UDP HEADER

S123-3 ATTACH FLAG

S123-4 ATTACH AUTHENTICATION VALUE & VERSION INFORMATION

S123-5 TRANSMIT TO CLIENT

END

Fig. 2d

START

S124 CLIENT RECEIVES

S125 UPDATE?

YES

NO

S125-1 WRITE VERSION INFORMATION

S125-2 WRITE UPDATE DATABASE

END